Enhanced recurrent neural network for short-term wind farm power output prediction

Article (Published Version)

Enhanced Recurrent Neural Network for Short-term Wind Farm Power Output Prediction.

E. C. Eze, C.R Chatwin
Department of Engineering & Informatics, University of Sussex. Brighton, U.K.
e.eze@sussex.ac.uk

Abstract.

Scientists, investors and policy makers have become aware of the importance of providing near accurate spatial estimates of renewable energies. This is why current studies show improvements in methodologies to provide more precise energy predictions. Wind energy is tied to weather patterns, which are irregular, especially in climates with erratic weather patterns. This can lead to errors in the predicted potentials. Therefore, recurrent neural networks (RNN) are exploited for enhanced wind-farm power output prediction. A model involving a combination of RNN regularization methods using dropout and long short-term memory (LSTM) is presented. In this model, the regularization scheme modifies and adapts to the stochastic nature of wind and is optimised for the wind farm power output (WFPO) prediction. This algorithm implements a dropout method to suit non-deterministic wind speed by applying LSTM to prevent RNN from overfitting. A demonstration for accuracy using the proposed method is performed on a 14-turbines wind farm. The model out performs the ARIMA model with up to 80% accuracy.

Keywords: Wind Power Output Prediction, Recurrent Neural Network, Deep learning, oLSTM, ARIMA, RMSE, MSE.

I Introduction.

The global energy report shows that power generation from wind rose to 54.6 gigawatts (GW) of installed capacity. China and the USA are leading with increasing installed capacity. Countries like Germany and India are showing a strong appetite for wind energy [1]. Wind data is stochastic; it is a very complex task to forecast the velocity of wind using linear approaches [2]. In addition, the length of the forecasting horizon has a correlation with the accuracy of forecasting methods. These horizons are ultra-short-term, short-term, medium and long-term. Ultra-short-term wind forecasting refers to wind data prediction in the range of a few minutes to one hour ahead [3]. Short-term counterparts are mainly for a period starting from one hour to several hours ahead, generally for unit commitment and operational security in the electricity market. Medium-term and long-term forecasting refers to longer time horizons [4]. Prediction of wind depends on several atmospheric factors, which strongly affect wind energy conversion. Effective operation depends on not only wind energy conversion but also power system reliability and load demand. Real power prediction from wind are classified into four main categories – persistence model, physical methods, statistical and artificial intelligence methods.

In the persistent method, the future wind speed is equivalent to the wind speed in the forecasting time [5]. This method is the most economical and the simplest wind forecasting approach. The drawback is the rapid degradation of the performance on an extended forecasting horizon. The physical approach however is based on numerical weather prediction (NWP) [6]. NWP outputs accurate estimations for long-term predictions. The major drawback of NWP models is the memory complexity and high time consumption in producing results; hence, it is not reliable for short forecasting horizons. The statistical methods find the mathematical relationship between wind-speed time series data. These models are auto regressive (AR), auto regressive moving average (ARMA), auto regressive integrated moving average (ARIMA), and Bayesian approaches. Reference [9] studied an approach using a K-nearest neighbour (KNN) regression model for short-term wind speed forecasting. Artificial intelligence (AI) techniques including artificial neural networks (ANNs) [7, 12-16], support vector regression (SVR) [17, 18], and recurrent neural methods [19-21] led to novel methodologies for wind speed and power predictions. ANNs can capture the relationships between the input data and the predicted wind speed values; hence, they are used for time series prediction of different weather variables on various time scales and yield. Recurrent ANN [20, 21], radial basis function (RBF) ANN [14] and adaptive wavelet ANN [22] have been proposed recently for wind speed prediction.

RNN-based approaches have been widely applied in the time series prediction. This is because RNN implicitly learns features in a high dimensional space applying its cell state capabilities. However, the drawback is that it suffers vanishing gradient challenges. Variants of RNN such as long short-term memory (LSTM) were proposed

[24, 25] to mitigate these problems but suffers overfitting especially on time series type of problems, hence, they require further control. Authors in [27] demonstrate the economic advantage of hybrid regularizations. In this paper, an integration involving fusing of regularization methods on RNN is proposed. This new regularization involves long short-term memory (LSTM) and a dropout regularisation (LSTM-Dropout) model for learning nonlinear temporal features from the wind series data to control overfitting.

The LSTM-Dropout (oLSTM) model is proposed to learn the decreasing energy function while increasing the learning pattern in the observed input vectors of the wind series dataset. The method effectively controls the vanishing gradient problem as it maps input and output wind data. The paper makes the following contributions:

- A new recurrent network-learning model, oLSTM, presented based on hybrid regularization of long short-term memory and dropout architectures for the robust unsupervised feature extraction of wind-series. The proposed oLSTM is an energy-based regressive method proven to capture the co-adaptation of input variables. To the best of our knowledge, oLSTM is the first recurrent deep learning model capable of capturing interval knowledge from wind data.
- The approach presented can extract meaningful features from the input in an unsupervised manner. Thus, unlike other AI methodologies [12-16], no prior knowledge about the wind data is needed for the feature extraction.
- In contrast to previous deep learning works including [29] and [25], this approach involves real-valued input units, designed for wind domain, which can work in any weather situation.

The contributions above are sub-divided into two areas: *a) Machine Learning:* The development of an integral long short-term learning system and the incorporation of the dropout tuning regularisation model and. *b) Wind farm Power output Prediction:* The application of an unsupervised feature extraction model in a nonlinear learning fashion to predict output wind power.

The rest of the paper is organised as follows; in Section II field data used in the research is discussed. Section III presents the implemented machine learning model configurations. The research experimentation is presented in section IV. However, the results evaluation and presentation is as discussed in section V while the paper conclusion is as presented in section VII.

II. Field Data Description

The data is extracted from the PHM society [30]. Feature extraction, de-noising and filtering were as described by [31]. Figure 1 is an exploration of randomly selected data within the wind farm.

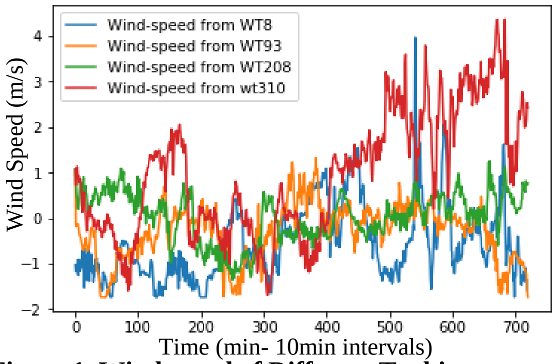

Figure 1. Wind speed of Different Turbines.

Visualisations at each of these Turbines, shows wind speed plots share similar patterns. In the research, wind data from fourteen turbines (although four are presented in the paper) is considered. These data were from wind turbine (WT330), wind turbine 291 (WT291), WT310 and so on.

Table 1: Summary Statistics of Wind farm Data.

	WT8	WT93	WT208	WT310
Count	720	720	720	720
Mean	0.534	0.356	0.283	0.145
Median	-0.008	-0.102	0.116	-0.161
Mode	-0.496	-0.320	0.230	-0.725
STD	1.485	1.046	0.898	1.021
S.Variance	2.204	1.094	0.807	1.041
Skewness	1.589	1.243	1.623	2.121
Range	7.984	4.704	4.372	5.261
C.Interval	1.109	0.077	0.066	0.075
Maximum	7.259	4.704	4.372	5.261

The description of Table 1 suggests either checking the variations statistically to ensure the merging of the associated data from different turbines is possible or using machine learning (ML) to generate individual predictions and merge results afterwards. However, the paper chose the former due to its reliability and efficient results.

III. Machine Learning Model Preparation

The paper employed these three basic steps to achieve data preparation prior to model fitting:

- Transform the generated wind speed (WS) data to be stationary using Dickey-Fuller test by computing first level (d = 1) differencing using the difference between current series (γ_t) and previous series (γ_{t-1}) as in $\Delta\gamma_t = \gamma_t - \gamma_{t-1}$.
- Transform the data into a supervised learning problem to have input/output patterns such that

at prior steps, observations are used as input to predict observation at the current time step
- Normalise the data to have a specific scale between -1 and 1.

These transforms are converted after the prediction to return them into their original scale before errors are calculated and scored.

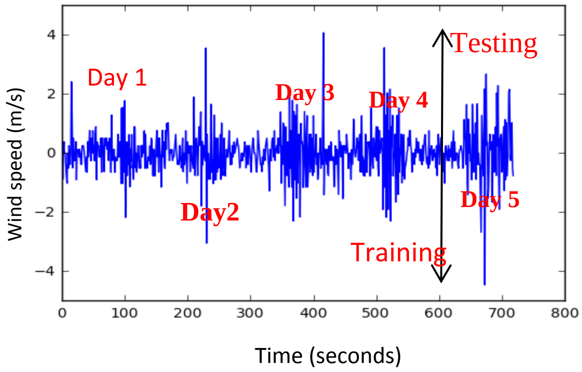

Figure 2: wind speed data at 10min intervals.

From the differenced data within the farm, Figure 2 implies that maximum wind speed is experienced between 4 A.M. to noontime leaving the afternoon time with low speed winds. The insight gained, led to data split; 80% of the data for training while the rest 20%, within the high wind time is set for testing the proposed model. The paper leveraged on the LSTM configuration and discussions for wind power output prediction.

a. LSTM Model Description.

First, the basic structure of LSTM is as shown in Figure 3 and the explanations of how the improvements are made over RNN are discussed in [25, 35, 36]. This Figure depicts the LSTM architecture with a single node cell implementation.

The cell state C_t updates input, i_t and output, o_t wind features by performing element-wise multiplication on the input and output gate of the cell while the previous state of the cell is multiplied by the forget gate f_t. This scenario results in the control of exponential bursts that corrects vanishing gradients. Because time series requires single value prediction, the gate activation function i_t, o_t, f_t *uses* sigmoid activation for LSTM output blocks. However, the rest of the mathematical configuration is as shown in Eq. (1).

$$\begin{aligned}
f_t &= \sigma_g(\Theta_{xf}x_t + \Theta_{hf}h_{t-1} + b_f)\\
i_t &= \sigma_g(\Theta_{xi}x_t + \Theta_{hi} + b_i)\\
o_t &= \sigma_g(\Theta_{xo}x_t + \Theta_{ho}h_{t-1} + b_o)\\
g_t &= Tanh\,(\Theta_{xg}x_t + \Theta_{hg}h_{t-1} + b_g)\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t\\
h_t &= o_t \odot \mathrm{Tanh}\,(c_t) \qquad (1)
\end{aligned}$$

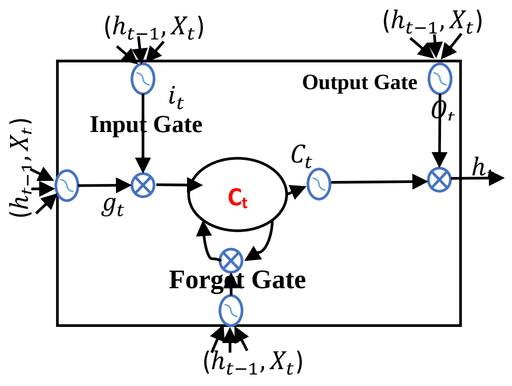

Figure 3: LSTM Architectural Design

b. Dropout Modelling.

This is another type of regularization method seen in ANN literature. However, in terms of implementation, authors in [37-39] applied dropout to solve various over-fitting challenges. Other methods [40] applied the Bernoulli random technique δ_i to remove certain hidden neurons í from a neural network having $P(\delta_i = 0) = q_i$ assumed to be independent to each other. $P(\delta i = 1) = 1 - q_i = p_i$ Linearity property ensures that the expectation of the output of the neuron is:

$$E[y^{(i)}] = \sum_{k=1}^{n} w_k x_k^{(i)} E[\delta_k] + bE[\delta_k]$$

$$= \sum_{k=1}^{n} w_k x_k^{(i)} p_k + bp_b \qquad (2)$$

From ii, $at\ IID,\ the\ random\ variable\ p_i = q\ and\ p_b = p$, result to

$$E[y^{(i)}] = \sum_{k=1}^{n} w_k x_k^{(i)} p + bp$$

$$= p.[\sum_{k=1}^{n} w_k x_k^{(i)} q + b] \qquad (3)$$

However, Eq. (3) results in a **dropout representation**, which is simplified to $\frac{1}{1-p} = \frac{1}{q}$ resulting in Eq. (4)

$$E[y^{(i)}] = \frac{1}{q} \cdot [\sum_{k=1}^{n} w_k x_k^{(i)} q + b] \qquad (4)$$

The equation further results in the concept of **dropout** that improves the generalisation of the LSTM neural configuration.

c. L1L2 Regularisation

Another technique used in improving RNN performance is weight regularisation. This technique imposes the L1L2 constraints on weights within LSTM nodes to reduce overfitting. Research in [24] mathematically resolves the idea in Eq. (5) below,

$$\mathcal{L}_T(w) = \mathcal{L}_D(w) + \lambda \mathcal{L}_w(w)$$

$$\mathcal{L}_D(w) = \frac{1}{2}\sum_{i=1}^{m}[y^{(i)} - h(x^{(i)}; w)]^2$$

$$\mathcal{L}_w(w) = \frac{1}{2} w^T w \quad (5)$$

Where $y^{(i)}$ is the target output while $h(x^{(i)}; w)$ is the network output, w is LSTM NN parameter and $\mathcal{L}_T(w)$ is a loss function associated with weight . $\boldsymbol{\lambda}$ is the regulariser that controls the trade-off between $\mathcal{L}_D(w)$ *and* $\mathcal{L}_w(w)$, while $\mathcal{L}_D(w)$ is the sum square error between $y^{(i)}$ *and* $h(x^{(i)})$

IV. Experiments

A. Model Training

The method for correcting weight values [41] to handle over-adaptation, which causes diminishing accuracy on new samples while training LSTMs is applied by the p in Eq. (3), which is multiplied by the weight parameters w_k of the LSTM network to present a fake neuron by reducing co-adaptation among the neurons. This scenario results in an LSTM network that is insensitive to specific neuron weights, thereby influencing better generalisation with relatively less likelihood for overfitting training data.

The oLSTM input wind-power pattern is framed into *samples, time steps and features*. Framing feature series is implemented using a window method that requires samples featured in current time (t) to predict the next time sequence (t+1) knowing prior times t-1, t-2, t-3,…, t-n as input variables. This is because from the literature, LSTM's gating parameters decides whether to update current state **ɱ** to new candidate state **Ш** learning from the input sequence of the previous state.

B. Optimisation criterion.

While training oLSTM, error is imminent. The algorithm experiences errors that need to be minimised. The error function **E(x)** depends on the internal learnable parameters of the model [41]. These learnable parameters are used to compute target values (**Y or predicted values**) from a set of **X or input wind features**. The weight (w_k) and bias (b_k) are learnable parameters that are used to compute output wind values that are learned and updated alongside the optimal solution in order words, having loss minimisation in the training process. During training, the paper implemented RMSprop as in [27].

C. ARIMA Configuration.

Machine learning (ML) models like LSTM, can be applied directly to the raw data [9], ARIMA $_{(P,\ D,\ Q)}$ models that are state space models requires model improvement due to outliers inherited from the data. The p d, and q parameters are modelled in [8] using the grid search machine learning method.

The grid search technique is slow, it depends on the performance of the system processor and RAM. It is tuneable to RMSE statistical quantity for best estimation. In this paper, about 0.82% evaluated RMSE error were reported meaning that the search has the best (p, d, q) components at (0, 2, 1) respectively.

D. Wind Power Modelling.

The set of inputs are multiplied by a set of weights (($w_{\theta i}$) and are further processed by individual deep units that have 12-hidden layers with output ϴ unit as in Eq. (6)

$$y_\theta(t) = P\left(f\left(oLSTM_\theta(t) and\ ARIMA_{p,d,q}\right)\right| \{ws_{t-1}, ws_{t-2,} ws_{t-3}, \dots\} \quad (6)$$

Where *ws = wind speed.*

$$oLSTM_\theta(t) = \sum_{i=1}^{\theta} w_{\theta i} X_i(t) \quad (7)$$

$$ARIMA_{p,d,q} = \sum_{i=1}^{\theta} X_i(t) \quad (8)$$

t represents 10 min interval of wind data. The sigmoid function implements as non-linear output of the form $(x) = \frac{1}{1+e^{-x}}$.

Modelling multi-step 8-hours ahead as proposed in Eq. (7) and (8) has N as the number of hours (minutes converted to hours) considered in the dataset. However, for each wind turbine at a particular hour, the formulation uses the generated weight as modified in Eq. (9).

$$W_d^{nod} = \{\dot{W}_s^{nod}[t_h + min], \dot{W}_{t,w,t,h}^{nod}[t_h + min]) \mid min = 1, 2, \dots, 14\} \quad (9)$$

$h = 1,2,3,\dots,8.$ $nod \in \{position\ of\ wind\ turbine\}$, which is not disclosed in the research dataset. $\dot{W}_{t,w,t,h}^{nod}$ *and* $\dot{W}_s^{nod}$ denotes turbine's node prediction of wind power respectively at time $t = t_h + min$

V. Evaluation and Result Presentation

The paper is evaluated by sequence generation using the mean squared error (MSE), which is averaged over the features in the training and test set. Comparing regularizers on RNN, the MSE on training and testing is as shown in Table 3. However, in Table 4 ARIMA and oLSTM predicted performance were compared using RMSE in different trials. Sequence generation on root mean squared error (RMSE) and MSE criterion is given by Eq. (10) and (11).

$$e_{MSE} = \frac{1}{N}\sqrt{\sum_{t=1}^{N}(X_t - \ddot{X}_t)} \quad (10)$$

$$ze_{RMSE} = sqrt(e_{MSE}) \quad (11)$$

Table 3: ARIMA, oLSTM Comparison.

	RMSE-ARIMA	RMSE oLSTM
@ 20%	0.7617	0.5283
@30%	0.8315	0.5052

For a better understanding[1] of each set of experiments, a total number of 10 samples is selected during training and testing with associated RMSE scores.

Table 4: Regularisation MSE results

Trials	oLSTM MSE (%)	L1L2 and LSTM MSE (%)	LSTM MSE (%)
Exp. 2	0.6240	0.6890	0.7102
Exp. 4	0.6731	0.6901	0.7806
Exp. 6	0.6224	0.6900	0.7412
Exp. 8	0.6105	0.7001	0.7510
Exp. 10	0.6001	0.7040	0.7306

The training and testing sample is evaluated after each training epoch to check if the configuration[2] is overfitting or under-fitting as in Figure 4.

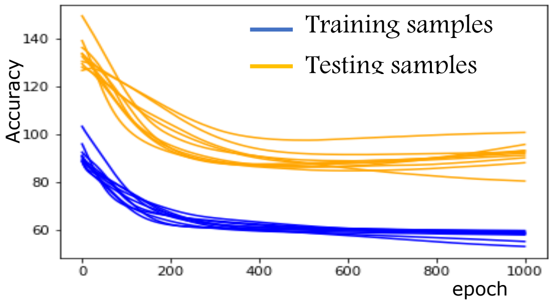

Figure 4: LSTM Overfitting Test

From the figure above, the model experienced closeness in the training sample, inferring overfitting. Overfitting is addressed by oLSTM as shown by the bumps observed in Figure 5. The boxplot of Figure 6 compared the distribution of results for each configuration.

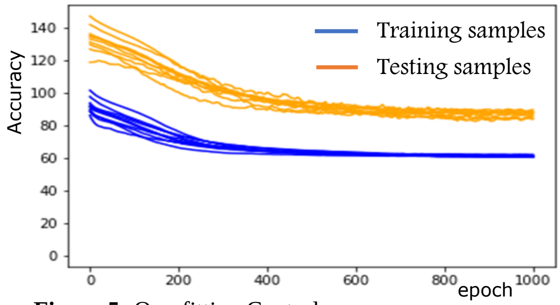

Figure 5: Overfitting Control.

Furthermore, LSTM experienced fast training with the type of regularizer used, while the rest – dropout and L1L2 appears to have good results with more hyper-parameter selection[3].

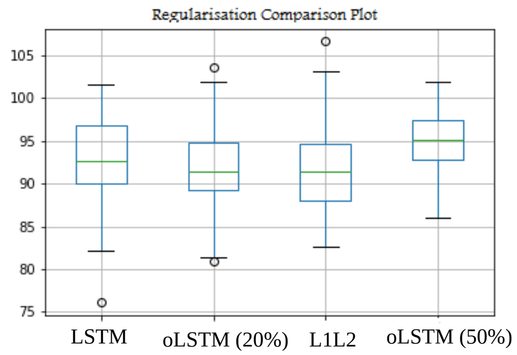

Figure 6: Regularisation Comparison.

Hence, looking at Figure 6, we infer that because the oLSTM has the least minimum median values, its generalisation shows a better model. Therefore, the model is used to derive Figure 7 and figure 8 that shows individual oLSTM and ARIMA models.

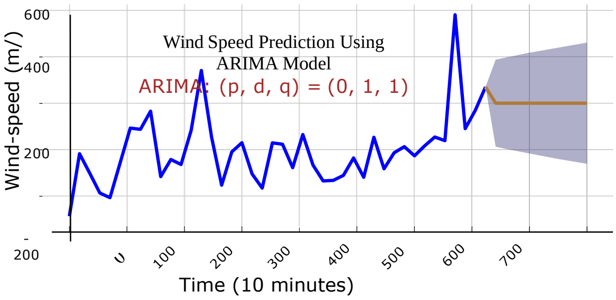

Figure 7: ARIMA wind power prediction.

[1] The dropout was set at 20% and L1L2 is of same sample set

[2] . Traditional LSTM with no regularisation

[3] LSTM/dropout and LSTM models are obtained with – learning rate: 2×10^{-3}, learning rate decay: 0.98, decay rate: 0.96, without dropout;

With the same setting, adding dropout to LSTM model has an adverse effect on its validation loss, similarly, when increasing the number of LSTM layers to 3. The number of layers are 2 for both RNN models, trained with a batch size of 10.

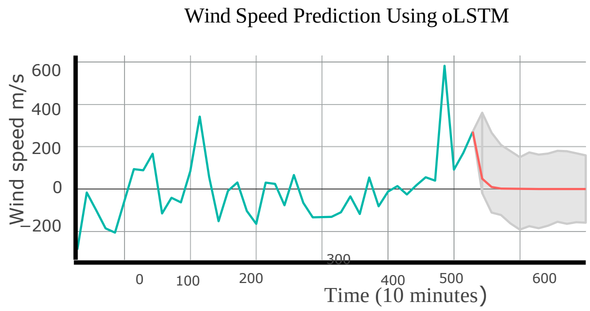

Figure 8: oLSTM wind-power prediction.

However, due to the performance of the dropout on LSTM, we compared the result with other machine learning algorithms – ARIMA of Figure 9 and 10.

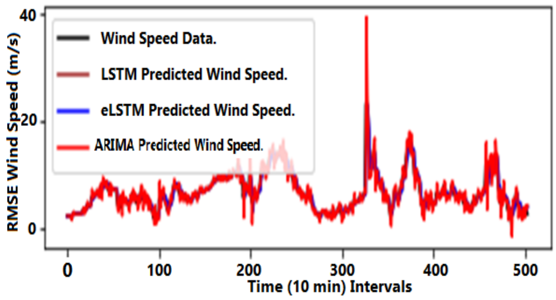

Figure 9: Model Comparison.

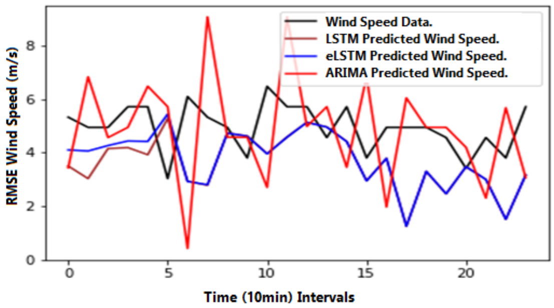

Figure 10: Model Prediction[4]

VI Conclusion

Wind speed prediction remains the most essential system variable for wind power predictions. Effective power system operation – load demand and penetration relies on accurate wind prediction. Important tools that improves efficiency and power system reliability depends on effective prediction of wind speed and power. This research paper has reviewed different techniques applied to wind speed and power. Techniques seen in wind speed forecasting differ from location to location, which in turn depends on the time of prediction. Several methods of prediction, application and metrics of performance used in studying different wind farms have been studied, especially the drawbacks of RNN, overfitting in Figure 5 and 6. Multi-step prediction, equivalent to short-term, eight-hour ahead wind-farm power output prediction using a new regularization method is carried out as depicted in Figure 7 and 8. In addition, wind speed prediction is considered for network training. The new strategy is used in selecting training samples for prediction of wind power based on wind speed.

References

[1] CNBC, "China and US lead way with wind power installations, says global energy report.," 05 May, 2017 13 Feb 2017

[2] C. Liang, P. Wang, X. Han, W. Qin, Y. Jia, and T. Yuan, "Battery Energy Storage Selection Based on a Novel Intermittent Wind Speed Model for Improving Power System Dynamic Reliability," *IEEE Transactions on Smart Grid*, 2017.

[3] Z. Guo, W. Zhao, H. Lu, and J. Wang, *Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model*. 2012, pp. 241–249.

[4] Z. Guo, W. Zhao, H. Lu, and J. Wang, "Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model," *Renewable Energy*, vol. 37, no. 1, pp. 241-249, 2012/01/01/ 2012.

[5] S. Balluff, J. Bendfeld, and S. Krauter, "Short term wind and energy prediction for offshore wind farms using neural networks," in *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*, 2015, pp. 379-382.

[6] T. Wardah, A. A. Kamil, A. B. S. Hamid, and W. W. I. Maisarah, "Statistical verification of numerical weather prediction models for quantitative precipitation forecast," in *2011 IEEE Colloquium on Humanities, Science and Engineering*, 2011, pp. 88-92.

[7] N. Chen, Z. Qian, I. T. Nabney, and X. Meng, "Wind Power Forecasts Using Gaussian Processes and Numerical Weather Prediction," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 656-665, 2014.

[8] M. Yesilbudak, S. Sagiroglu, and I. Colak, "A new approach to very short term wind speed prediction using k-nearest neighbor classification," *Energy Conversion and Management*, vol. 69, pp. 77-86, 2013/05/01/ 2013.

[10] H. Liu, H.-q. Tian, and Y.-f. Li, "Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for

[4] For ease of clarity, the figure was presented Reducing the test on 20 test samples

wind speed prediction," *Applied Energy*, vol. 98, pp. 415-424, 2012/10/01/ 2012.

[11] S. Baran, "Probabilistic wind speed forecasting using Bayesian model averaging with truncated normal components," *Computational Statistics & Data Analysis*, vol. 75, pp. 227-238, 2014/07/01/ 2014.

[12] D. Lee and R. Baldick, "Short-Term Wind Power Ensemble Prediction Based on Gaussian Processes and Neural Networks," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 501-510, 2014.

[13] H.-z. Li, S. Guo, C.-j. Li, and J.-q. Sun, "A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm," *Knowledge-Based Systems*, vol. 37, pp. 378-387, 2013/01/01/ 2013.

[14] W. Zhang, J. Wang, J. Wang, Z. Zhao, and M. Tian, "Short-term wind speed forecasting based on a hybrid model," *Applied Soft Computing*, vol. 13, no. 7, pp. 3225-3233, 2013/07/01/ 2013.

[15] J. Shi, Z. Ding, W.-J. Lee, Y. Yang, Y. Liu, and M. Zhang, "Hybrid forecasting model for very-short term wind power forecasting based on grey relational analysis and wind speed distribution features," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 521-526, 2014.

[16] S. Haykin, *Kalman filtering and neural networks*. John Wiley & Sons, 2004.

[17] S. Salcedo-Sanz, E. G. Ortiz-Garcı´a, Á. M. Pérez-Bellido, A. Portilla-Figueras, and L. Prieto, "Short term wind speed prediction based on evolutionary support vector regression algorithms," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4052-4057, 2011/04/01/ 2011.

[18] J. Liu and E. Zio, "SVM hyperparameters tuning for recursive multi-step-ahead prediction," (in English), *Neural Computing & Applications*, Article vol. 28, no. 12, pp. 3749-3763, Dec 2017.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[20] O. Wallscheid, W. Kirchgässner, and J. Böcker, "Investigation of long short-term memory networks to temperature prediction for permanent magnet synchronous motors," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1940-1947.

[21] W. Yao, P. Huang, and Z. Jia, "Multidimensional LSTM Networks to Predict Wind Speed," in *2018 37th Chinese Control Conference (CCC)*, 2018, pp. 7493-7497.

[22] B. Kanna and S. N. Singh, "Long term wind power forecast using adaptive wavelet neural network," in *2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*, 2016, pp. 671-676.

[23] D. T. Mirikitani and N. Nikolaev, "Recursive Bayesian Recurrent Neural Networks for Time-Series Modeling," *IEEE Transactions on Neural Networks*, vol. 21, no. 2, pp. 262-274, 2010.

[24] J. Brownlee, "How to Use Weight Regularization with LSTM Networks for Time Series Forecasting," p. 1, May 5, 2017 2017.

[25] M. Coto-Jimenez and J. Goddard-Close, "LSTM Deep Neural Networks Postfiltering for Enhancing Synthetic Voices," (in English), *International Journal of Pattern Recognition and Artificial Intelligence*, Article; Proceedings Paper vol. 32, no. 1, p. 24, Jan 2018, Art. no. 1860008.

[26] D. Q. Wang, Y. L. Gao, J. X. Liu, C. H. Zheng, and X. Z. Kong, "Identifying drug-pathway association pairs based on L1L2,1-integrative penalized matrix decomposition," (in English), *Oncotarget*, Article vol. 8, no. 29, pp. 48075-48085, Jul 2017.

[27] V. Pham, C. Kermorvant, and J. Louradour, *Dropout Improves Recurrent Neural Networks for Handwriting Recognition*. 2013.

[28] T. Moon, H. Choi, H. Lee, and I. Song, *RnnDrop: a novel dropout for RNNs in ASR*. 2015, pp. 65-70.

[29] T. Bluche, C. Kermorvant, and J. Louradour, "Where to apply dropout in recurrent neural networks for handwriting recognition?," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 681-685.

[30] M. Fei and D. Y. Yeung, "Temporal Models for Predicting Student Dropout in Massive Open Online Courses," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 256-263.

[31] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83-95, 2016.

[32] P. Society, "phm11 data challenge - condition monitoring of anemometers," https://www.phmsociety.org/competition/p

mh/11/problems. , Case study 10/11/2014 2011.

[33] L. Sun, C. Chen, and Q. Cheng, "Feature extraction and pattern identification for anemometer condition diagnosis," *Int. J. Progn. Heal. Manag*, vol. 3, pp. 8-18, 2012.

[34] D. Kececioglu, *Reliability engineering handbook*. DEStech Publications, Inc, 2002.

[35] R. Maalej and M. Kherallah, "Improving MDLSTM for Offline Arabic Handwriting Recognition Using Dropout at Different Positions," in *Artificial Neural Networks and Machine Learning – ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II*, A. E. P. Villa, P. Masulli, and A. J. Pons Rivero, Eds. Cham: Springer International Publishing, 2016, pp. 431-438.

[36] Y. Fei, J. Hu, K. Gao, J. Tu, W.-q. Li, and W. Wang, "Predicting risk for portal vein thrombosis in acute pancreatitis patients: A comparison of radical basis function artificial neural network and logistic regression models," *Journal of critical care*, vol. 39, pp. 115-123, 2017.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[38] G. Vaca-Castano, S. Das, J. P. Sousa, N. D. Lobo, and M. Shah, "Improved scene identification and object detection on egocentric vision of daily activities," *Computer Vision and Image Understanding*, vol. 156, pp. 92-103, 3// 2017.

[39] S. V. Ravuri and A. Stolcke, "Recurrent neural network and LSTM models for lexical utterance classification," in *INTERSPEECH*, 2015, pp. 135-139.

[40] T. N. Sainath *et al.*, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, pp. 39-48, 2015.

[41] E. Phaisangittisagul, "An Analysis of the Regularization between L2 and Dropout in Single Hidden Layer Neural Network," *IEEE Computer society*, Conference Paper vol. DOI 10.1109/ISMS.2016.14, no. 2166-0670/16, p. 6, 2016 2016.